:VALID *TDB-ACC-NO: NN9212203

DISCLOSURE TITLE: Principles for Designing a Series of Three Dimensional Graphics Adapters.

PUBLICATION-DATA: IBM Technical Disclosure Bulletin, December 1992, US

VOLUME NUMBER: 35

ISSUE NUMBER: 7

PAGE NUMBER: 203 - 206

PUBLICATION-DATE: December 1, 1992 (19921201)

CROSS REFERENCE: 0018-8689-35-7-203

DISCLOSURE TEXT:

- A systematic framework for designing a family of graphics adapters for a computer workstation is disclosed. A graphics adapter is a set of electronic circuits which produce analog video signals under the control of a host processor. The analog video signals produce a rectangular pattern of dots or pixels on a cathode ray tube (CRT). Although several specific graphics adapter designs are presented as parts of this invention, the principal purpose of this invention is to establish the systematic relationships between these graphics adapters.

Description of the apparatus

The apparatus resulting from this invention consists of a set of graphics adapters. Each of these adapters includes the following basic components:

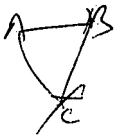
1. A printed circuit board providing physical support for the remaining components as well as supporting appropriate electrical connections between these components and the host bus.
2. A bus interface circuit (BIC) providing electrical connection between the host bus and subsequent components identified below.
3. A frame buffer (FB) consisting of digital memory in sufficient quantity to record all of the information in a rectangular block of dots or pixels.
4. Digital to analog converters (DACs), capable of reading the frame buffer memory and producing analog video signals for refreshing the screen of a CRT.
5. Pixel storage unit (PSU). This is a circuit capable of taking pixel data values from the bus interface circuit and storing them in appropriate addresses within the frame buffer memory.

- In addition to these basic parts, each graphics adapter contains one or more of the following components:

1. Depth buffer (Z-buff). This circuit contains memory sufficient for storing a depth value (Z) for each pixel value defined in the frame buffer memory.
2. Scan line processor (SLP). This

scan-line data as recognized by the scan-line processor.
4. Triangle set-up processor (TSP). This circuit must be capable of transferring pixel data, scan-line data, or triangle area fill data from the bus interface circuit to a triangle area fill processor unchanged. In addition, this circuit must be capable of receiving data representing a sequence of triangles defined by a set of vertex data records. Each vertex data record contains the coordinates (x, y, z) and color values (r, g, b) for a 3d point. The first three vertex data records define the first triangle in a sequence.

Each subsequent vertex data record defines a new triangle by replacing one of the three vertices in the previous triangle, exclusive of the most recent previous vertex. This circuit must use the values for each triangle to calculate the parameters required by the triangle area fill processor. The resulting parameters must then be transferred to the triangle area fill processor.



- These components may be combined in the following four configurations:

1. Adapter 1: Basic parts only.
2. Adapter 2: Basic parts plus scan line processor and depth buffer.
3. Adapter 3: Basic parts plus triangle area fill processor, depth buffer, and one or more sets of scan-line processors and pixel storage units.
4. Adapter 4: Basic parts plus triangle set-up processor, triangle area fill processor, depth buffer, and one or more sets of scan-line processors and pixel storage units.

Operation of the invention

This invention is based on the use of the following algorithm to convert a sequence of triangles into a set of dots (pixels) on a CRT. This sequence of triangles may represent a convex polygon, a triangle strip, or other drawing primitives defining a 3d surface area.

1. Deliver 3d vertex data to the graphics system.
2. Convert the vertex data into colored vertex data in device coordinates.
3. Transfer colored vertex data to triangle set-up processor, if present.
4. Form triangles by combining each vertex with two of the three vertices in the preceding triangle.
5. Transfer the results of the triangle set-up processing to the scan line processor, if present.
6. Generate scan line parameters for each triangle, based on the Gonzalez-Liang triangle set-up parameters calculated by the triangle set-up processor.
7. Transfer scan-line data to the scan-line processor, if present.
8. Convert scan lines into z-buffered pixels.
9. Read z-buffer values, write new depth values to the z-buffer, and transfer the masked pixel sequences to the pixel storage unit.

- This algorithm is executed on each of the four graphics adapters identified in the section on the apparatus. In the case of Adapter 1, steps 1 through 8 execute on the host CPU and step 9 is performed using the host bus as the communications pathway from the host to the adapter. In this case, the maximum rate at which triangles can be processed by the ho

bus bandwidth divided by 79 bytes.

- In the case of Adapter 4, steps 1 and 2 are executed on the host processor, step 3 occurs at the host bus, and steps 4 through 9 are performed on the graphics adapter. In this case, the maximum rate at which triangles may be processed is no greater than the host bus bandwidth divided by 10 bytes.
- In each case, the maximum triangle processing rate is directly proportional to the host bus bandwidth. The following material is based on a host bus bandwidth of 27.6 Mbytes/sec because this is the rate required to display images with a resolution of 640x480x24 bits at a rate of 30 frames per second. The actual bandwidth of the host bus in a particular system may be higher or lower than this value. The performance figures presented below may be adapted to any specific host bus bandwidth by multiplying by the ratio of the actual bandwidth to 27.6 Mbytes/sec. The performance requirements and capabilities of each of these types of graphics adapters are summarized in Table 1.

For example, in the case of Adapter #2, the host processor must be capable of delivering 10.6 Mflops of floating-point processing power, the adapter must be capable of storing 5.3 M pixels/sec into the frame buffer, and the resulting system would be capable of rendering 106 k triangles per second.

***** SEE ORIGINAL DOCUMENT *****

Table 1. Performance requirements for Graphics Adapters #1 to #4.						
Stage	Load/vertex	#1	#2	#3	#4	units
1	24 bytes	1.25	2.5	10.4	66.4	Mbyte/sec
2	100 instr	5.2	10.6	43.2	276	Mips
2	100 flops	5.2	10.6	43.2	276	Mflops
3	10 bytes	(0.5)	(1.1)	(4.3)	27.6	Mbyte/sec
4	100 instr	5.2	10.6	43.2	276	Mips
5	64 bytes	(3.3)	(6.8)	27.6	177	Mbyte/sec
6	200 instr	10.4	21.2	86.4	552	Mips
7	261 bytes	(13.6)	27.6	113	720	Mbyte/sec
8	800 instr	41.6	84.8	346	2208	Mips
9	530 bytes	27.6	58.7	230	1463	Mbyte/sec

vertices/sec	52k	106k	432k	2.8M
triangles/sec	52k	106k	432k	2.8M
host Mflops	5.2	10.6	43.2	276
adapter Mflops	0	0	0	0
host Mips	62	42	86	276
adapter Mips	0	85	100	3036
M pixels/sec	2.6	5.3	22	138